

Learned Inference of Annual Ring Pattern of Solid Wood

Maria Larsson^{1†} Takashi Ijiri^{2†} I-Chao Shen³ Hironori Yoshida⁴ Ariel Shamir⁵ Takeo Igarashi⁶

¹ ma.ka.larsson@gmail.com, The University of Tokyo

² takashi.ijiri80@gmail.com, Shibaura Institute of Technology

³ ichaoshen@g.ecc.u-tokyo.ac.jp, The University of Tokyo

⁴ hironoriyh@gmail.com, Future University Hakodate

⁵ arik@runi.ac.il, Reichman University

⁶ takeo@acm.org, The University of Tokyo

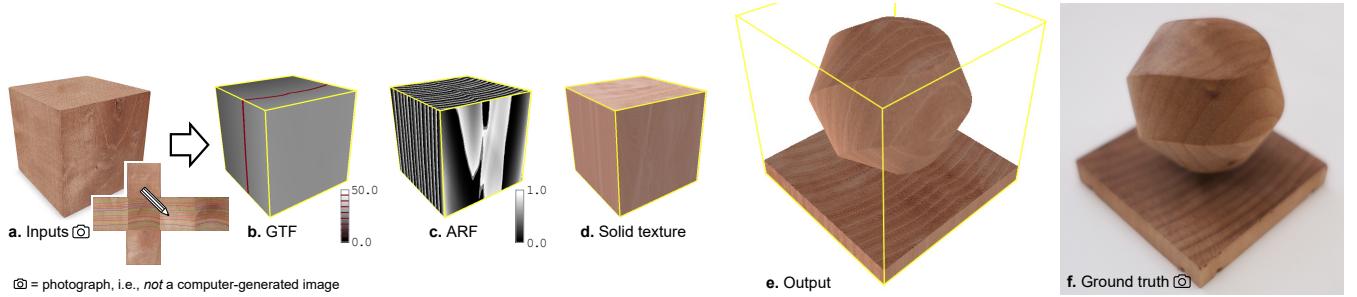


Figure 1: a) Photographs of a wood block and its unfolded external surfaces with manually traced annual rings. b) The inferred volumetric growth time field (GTF). c) The annual ring field (ARF) converted from the GTF. d) A solid wood texture with realistic colors and detailed features. e) A rendered cut surface inside the volume. f) The corresponding cut of the physical wood block.

Abstract

We propose a method for inferring the internal anisotropic volumetric texture of a given wood block from annotated photographs of its external surfaces. The global structure of the annual ring pattern is represented using a continuous spatial scalar field referred to as the growth time field (GTF). First, we train a generic neural model that can represent various GTFs using procedurally generated training data. Next, we fit the generic model to the GTF of a given wood block based on surface annotations. Finally, we convert the GTF to an annual ring field (ARF) revealing the layered pattern and apply neural style transfer to render orientation-dependent small-scale features and colors on a cut surface. We show rendered results of various physically cut real wood samples. Our method has physical and virtual applications such as cut-preview before subtractive fabricating solid wood artifacts and simulating object breaking.

CCS Concepts

• Computing methodologies → Volumetric models;

1. Introduction

Solid wood is a ubiquitous material with a characteristic texture that is unique to each material instance. Its modeling includes two aspects: 1) the global structure of the annual ring pattern and 2) the detailed appearance of the color transitions and small-scale features. In this paper, we address these two aspects aiming to

infer a solid texture based on the visible exterior surfaces of a physical block of wood (Figure 2). This is potentially useful for physical as well as virtual applications. When it comes to physical applications, it enables predicting the appearance of wood artifacts before fabrication by subtractive manufacturing, such as milling or cutting. Regarding virtual applications, it enables the generation of the texture of a surface that is revealed when an object breaks. The inputs to our method are photographs of the six external surfaces of a wood block and annotations on these photographs in the form

† These two authors contributed equally to this work.

of traces of the annual rings numbered by year. The output is a volumetric model of the wood block enabling the rendering of a cut surface inside the block.

For the first aspect of wood texture modeling—1) the global structure of the annual ring pattern—our key insight is that the pattern can be represented akin to a signed distance field (SDF), which is typically used for implicit surface modeling. This similarity in representation is significant because it opens up the possibility to leverage SDF-based 3D generative modeling techniques for our problem. Specifically, Park et al. proposed a network for SDF modeling of 3D geometries capable of shape completion and interpolation from noisy and partial input 3D data [PFS*19]. We employ this network to a different domain—volumetric material modeling—aiming to complete annual ring patterns based on partial information in the form of the ring traces on the exterior surfaces.

The SDF-like data representation that we use is referred to as a growth time field (GTF). It is based on the logic of cambial tree growth, which is the type of growth that gives rise to the annual ring pattern. Cambial growth starts from the center and proceeds outwards, consecutively adding a layer of material over the previous layer each year. We associate the cylindrical surface of each layer with the growth time (year), obtaining a 3D continuous field—the GTF. It is nearly equivalent to a distance field around the pith (the central skeleton of a tree); the difference is that there is a degree of natural distortion in the GTF caused by variations of speed of growth at different points (otherwise the annual rings would be perfectly circular). This representation has been used for forward modeling of tree growth and annual ring patterns [MPW06, SPH11, KSG*15, LIY*22]. However, we address the backward problem of inferring the internal GTF from external surfaces.

To achieve the inference, we first train a generic neural model (based on the work of Park et al. [PFS*19]) of GTFs that predicts the GTF value at each 3D location of a wood cuboid. This model is trained with many procedurally generated GTF samples. Then, we optimize a latent vector and a cutout transformation such that it matches the annotated ring pixels on the six sides of the wood block as closely as possible. After obtaining the GTF, it is trivial to convert it to an annual ring field (ARF) by a modulus-based operation, visualizing the layered pattern. An ARF is defined as a field ranging from 0.0 to 1.0, where a lower value indicates earlywood (grows in the spring, typically has a lighter color in real wood) and a higher value indicates latewood (grows in the fall, typically darker). The output of the first stage is thus a volumetric abstract representation of the global structure (the ARF).

In the next stage—2) local appearance synthesis—we address the problem of applying realistic colors and detailed features to the global annual ring structure inferred in the previous stage. We achieve this by neural style transfer, i.e., combining the content of one image with the “artistic style” of another. In our case, the content image is the global structure (the ARF) and the “artistic style” is the original color photographs of the exterior of the wood block. Specifically, we apply an existing patch-based style transfer method [TFK*20] that can be trained on just one image pair (an ARF image and a color photograph). This allows us to re-train the model for the specific style of each wood exemplar. After training, the style transfer is applied to each layer of the volume at a given

resolution to achieve a realistic color appearance to the full interior volume of the wood block.

Moreover, in the real world, the local appearance of wood material varies depending on in which plane the wood is cut. The end grain—where fibers and pores are facing the surface—tends to be rougher and slightly darker compared to the face grain, where fibers and pores are tangential to the surface (Figure 3). For the purpose of modeling this anisotropic appearance, we perform the style transfers on three orthogonal cross-sections (the exterior surfaces facing the in x- y- and z-directions) and render a point on a cut surface based on its surface normal. If the normal of a point faces towards the end-grain, the proportion of color contribution from the end-grain image is increased, and so on. The final result of the combined global structure inference and local appearance synthesis is a volumetric solid wood texture corresponding to a physical sample, which can be used to render a cut surface inside the volume with a realistic appearance of the global structure and anisotropic local features.

We evaluate our method on several physical solid wood samples by inferring their volumetric textures, after which we cut the samples to reveal their interior textures. We present these ground truth cut surface textures side-by-side with the corresponding predicted textures for qualitative visual comparison. We also present a quantitative ablation and baseline comparison study. Although not statistically significant, the study seems to suggest that our method more robustly infers internal structures from partial annotations compared to naïve interpolation using the radial basis function (RBF). Furthermore, we show additional visual results comparing orientation-dependent and orientation-agnostic rendering. We also show the effect of changing the orientation of a cut surface inside a wood block and cross-combinations of global structures with different local appearances.

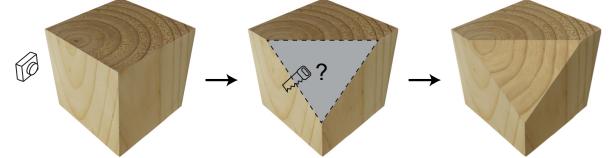


Figure 2: We aim to infer the internal volumetric texture of solid wood given photographs of the exterior surfaces.

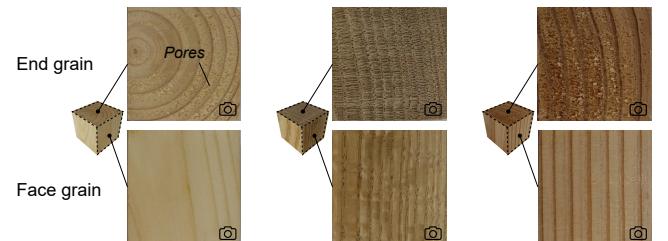


Figure 3: Photographs of different sides of real material samples. The local appearances (colors, small-scale features of pores, etc.) varies on cross-sections in different orientations.

In summary, our contributions are twofold:

- A method for inferring the global volumetric structure of a wood block based on annotated annual rings on the external surface using a learned generic model of annual ring patterns.
- An orientation-dependent rendering method for visualizing a cut surface inside a solid wood block considering its anisotropic nature.

2. Related Works

2.1. Tree Growth and Annual Ring Pattern Formation

Tree growth is divided into two types: *apical* and *cambial*. Apical growth elongates the tree and produces new grafting strands, and it is typically modeled with L-systems [GJB^{*}20, Lin68, IOI06]. Cambial growth expands the thickness of the tree by adding a layer of material to its external surface each year, and it is typically simulated using the level set method [KSG^{*}15, MPW06]. This second type of expansion growth gives rise to the annual ring pattern. In temperate climates, wood grows fast in the spring, adding a thick layer of light material. In the fall, the growth slows down, adding a thin layer of dark material, until stopping completely in the winter. These seasonal shifts in color make the growth layers, i.e. annual rings, visible to the eye. Moreover, various factors (e.g., tree species, sunlight, nutrition) give rise to variations in color and thickness. Therefore, each piece of wood has a unique volumetric texture, even two pieces from the same species are never the same.

2.2. Procedural Wood Texturing

Procedural texturing is a method to generate a texture on demand based on mathematical functions and rules, rather than referring to a stored raster image. Based on this approach, Liu et al. [LDHM16] proposed a simulation method to generate a volumetric structure and texture of solid wood. With their method, a user can control various visual features of wood by tuning parameters, such as the distance between annual rings, colors, surface reflectance, and distortions. Moreover, Larsson et al. [LIY^{*}22] proposed a method to procedurally model knots and the intricate annual ring distortions they give rise to. These procedural methods are computationally efficient and suitable for many applications, such as in games and videos where the goal is to make the texture look plausible. However, these are forward methods and do not refer to specific exemplars.

The inference of parameters and generation of procedural models is an active research area [LP00, GHG^{*}22, HHD^{*}22, HDR19, GHS^{*}22]. For the highly realistic rendering of wood material, Marschner et al. [MWAM05] presented a shading model dealing with sub-surface highlighting specific to finished wood and a measurement method to obtain rendering parameters from wood samples. Lefebvre and Poulin [LP00] modeled the structure of wood; specifically, introducing a procedural wood model with concentric circular annual rings and extracting its parameters from a 2D image. Our method is different in that we separate the structure estimation and appearance synthesis, provide a more general annual ring model, and adopt the style transfer for the appearance synthesis. Moreover, we chose a neural network over a procedural framework for the

inference problem because a neural network is differentiable, making it easy to fit, while it can be difficult to directly fit an explicit procedural model to a given input.

2.3. Solid Texture Synthesis

Another approach to model volumetric materials, including wood is to synthesize a 3D texture from a 2D reference image using non-parametric sampling or deep learning. Kopf et al. [KFCO^{*}07] synthesized volumetric textures from single 2D images. They extended a 2D texture optimization method [KEBK05] to 3D and integrated a histogram matching technique with the texture optimization procedure. Dong et al. [DLTD08] synthesized only parts of the volume used for rendering and performed pre-computation to reduce the candidates of neighborhood matching. They also generated anisotropic volumes from multiple 2D images. Pietroni et al. [POB^{*}07] synthesized a volumetric texture from multiple cross-sectional images based on a morphing technique. For more related studies, refer to the survey paper by Pietroni et al. [PCOS10].

Recently, deep neural network models have been applied to solid texture synthesis. Gutierrez et al. [GRGH18] and Zhao et al. [ZWG^{*}21] adopted generative adversarial networks to synthesize solid textures from 2D exemplar images. They introduced multi-scale representation to their models. Gutierrez et al. [GRGH18] evaluated the similarity between the given exemplars and generated volumes using their perceptual feature vectors of the images. In contrast, Zhao et al. [ZWG^{*}21] compared 2D patches extracted from the exemplar and generated volume directly without feature extraction. Furthermore, Oechsle et al. [OMN^{*}19] represented a 3D texture using a multilayer perceptron (MLP) model that maps 3D position to appearance.

Henzler et al. [HMR19] reconstruct diverse captured textures with infinite zoom by mapping them to latent texture codes and synthesize them using a MLP model with Perlin noise. We applied their model (a pre-trained version provided by the authors) to our captured images of solid wood and observed that the reconstructed output textures have a low resemblance to the target inputs (Figure 4). However, note that direct comparison between our method and that of Henzler et al. is not appropriate because their network takes different inputs and outputs. Most notably, our method requires the input of manual annotations. Moreover, the training data is different: we train on procedurally generated 3D data whereas they train on 2D images.



Figure 4: Results of applying the pre-trained texture synthesis model provided by Henzler et al. [HMR19] to our captured images of solid wood. Their model is material-specific and was trained on a non-public wood dataset.

Furthermore, Portenier et al. proposed a similar noise-based deep neural network (DNN) model [PABG20]. These methods can synthesize solid textures from 2D exemplars. However, they focus on

local appearances and are therefore most suitable for globally uniform materials, such as grass and gravel. They do not reproduce the global structure of wood, i.e., the concentric annual ring pattern.

2.4. Neural Surface Modeling

Neural implicit surface modeling is a technique to represent 3D shapes or surfaces implicitly using neural networks. Specifically, MLP networks define a scalar function $f(\mathbf{z}, \mathbf{x}) \in R$ that maps a latent vector \mathbf{z} and 3D position \mathbf{x} to a scalar value and where R is a set of real numbers. By computing the zero-level set of this function, a 3D surface model can be obtained. The versatility of neural implicit modeling allows for the generation of diverse models by manipulating the latent vector input to the MLP network. Various methods employed encoder networks to compute the latent vector [MON*19, CZ19, XWC*19, JSM*20]. Meanwhile, Park et al. [PFS*19] proposed a method to train the distribution of latent vectors from the dataset using auto-decoder-style training. Based on this neural implicit modeling technique proposed by Park et al. [PFS*19], we extend its application to modeling a volumetric material, moving beyond its conventional usage in single-view 3D reconstruction and mesh reconstruction from point clouds.

A Neural Radiance Field (NeRF) is another highly effective representation that implicitly models the surface geometry and appearance of a shape [MRS*20, MST*20]. However, it focuses on capturing the appearance of the exterior surfaces, rather than the inside of solid materials.

3. Method

3.1. Overview

Given the exterior surface images of an orthogonal wood block with annotated annual rings, our goal is to obtain a representation that recovers its internal structure and renders a textured image of a cut surface. We propose a two-stage method (Figure 5). In the first stage, we infer the global structure of the annual ring pattern (Section 3.2). We represent the internal global structure by a GTF. We train a generic neural model to predict the GTF value at each 3D location of a wood cuboid. A predicted GTF is then converted into an ARF to obtain a volume closer in structure to the actual layered appearance of wood. In the second stage, we perform local texture synthesis (Section 3.3). Specifically, we generate a style transfer model for each of three axis orientations and construct three RGB volumes by applying the style transfer models to the inferred ARF volume. Then, we combine the three RGB volumes to render a cut surface.

3.2. Global Structure Inference

3.2.1. Generic Neural GTF Model

In the proposed method, we represent a volumetric annual ring pattern using the GTF

$$GTF(\mathbf{p}) = g : \forall \mathbf{p} \in \mathcal{R}^3, g \in \mathcal{R}, \quad (1)$$

where \mathbf{p} is a 3D point inside a block of wood with its pith axis aligned with the z -axis. The GTF value (g) represents the time (year) when growth occurred relative to the innermost center (pith). That is

to say, the value of g along the pith is 0 and it increases as the growth expands outwards from the pith. We represent a generic model of GTFs using an MLP neural network:

$$f_\theta(\mathbf{c}, \mathbf{p}) \approx GTF(\mathbf{p}), \quad (2)$$

where θ represents the network parameters and \mathbf{c} is a latent code. The generic model covers a wide range of GTFs (variations in annual ring patterns) using the latent code to embed the variation in GTFs in a latent space. We inherit the MLP network architecture from Park et al. [PFS*19] who developed it for implicit surface modeling via signed distance fields (SDFs), which are similar to GTFs.

We choose the GTF as representation because of its similarity with an SDF, and the documented interpolation ability of the 3D generative SDF modeling network proposed by Park et al. [PFS*19]. Moreover, we reason that learning a GTF is more preferred over learning a ARF directly because the GTF contains global information while the ARF contains only local information. Moreover, ARF generated from a smooth GTF is guaranteed to be consistent (year rings are always closed), which would not be guaranteed in directly generated ARFs.

3.2.2. Training the Generic Neural GTF Model

During training, we optimize the network parameters θ of our model together with latent codes $\mathbf{c}_i (i = 0, \dots, N)$ for each $GTF_i (i = 0, \dots, N)$ in a procedurally generated training dataset (Section 3.2.5). Given N GTFs, we sample M points from each GTF to obtain pairs $\{(\mathbf{p}_j^i, g_j^i) | i \in (0, 1, \dots, N-1), j \in (0, 1, \dots, M-1)\}$, where $\mathbf{p}_j^i \in \mathbb{R}^3$ is the 3D position of the j th sample point of GTF_i , and g_j^i is its growth time value. Using the sampled training pairs, we optimize $C = \{\mathbf{c}_i\}$ and θ as follows:

$$\arg \min C, \theta \sum_{(i,j)} ||f_\theta(\mathbf{c}_i, \gamma(\mathbf{p}_j^i)) - g_j^i||_1 + w_{\text{reg}} \|\mathbf{c}_i\|_2 \quad (3)$$

where the first term penalizes the deviation of the predicted GTF values from the known GTF values, while the second term regularizes the distribution of latent vectors. To represent the concentric structure of annual ring patterns, we apply a cylindrical positional encoding, which uses a mapping function γ :

$$\gamma(\mathbf{p}) = (r, \arccos(x/r), \arcsin(y/r), z), \quad (4)$$

where $r = \sqrt{x^2 + y^2}$. We set the coefficient $\alpha = 0.01^2$, initialize $\mathbf{z}_i \in \mathbb{R}^{50}$ by sampling from $\mathcal{N}(0, 1/\sqrt{50})$ [PFS*19], and use the Adam optimizer [KB14]. This auto-decoder-style training produces high-quality generative models without requiring an encoder module.

3.2.3. Inferring a GTF

Given six surface images of a wood block with annotated annual ring pixels and their growth time values, our goal is to infer the volumetric GTF that best matches the annotations. We denote the annotated annual ring pixels as (\mathbf{x}_k, g_k) , where $\mathbf{x}_k \in \mathbb{R}^3$ is the location on the external surface of the wood block, and g_k is the growth time value of \mathbf{x}_k . When the pith (innermost tree center) appears on the exterior surface, we set $g_k = 1$ at the innermost annual ring and increment the age from there. When the pith does not appear, we set a tentative value (e.g., $g_k = 10$) for the youngest ring. The age offset is optimized afterwards (refer to Equation 5).

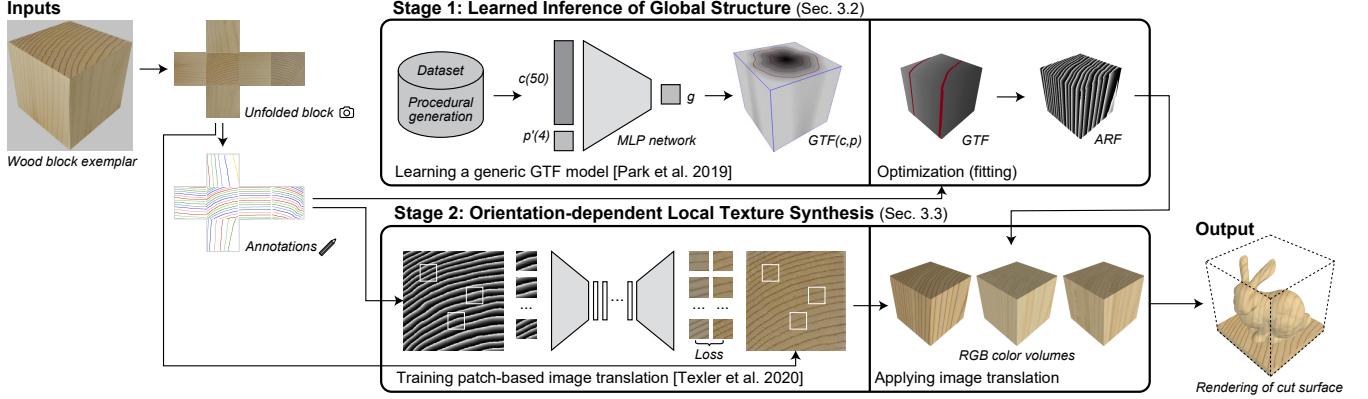


Figure 5: Method overview. The inputs are the external photographs of a wood block annotated with annual ring traces. The method consists of two stages: 1) learned inference of the global structure of annual rings and 2) orientation-dependent local texture synthesis. The output is a rendering of a cut surface inside the wood block.

The inputs to the generic model are a point (\mathbf{p}_k) and a latent vector (\mathbf{c}). The output is the growth time ($f_\theta(\mathbf{c}, \mathbf{p}_k)$) at the point (\mathbf{p}_k). Intuitively, we obtain a GTF for a given wood block by optimizing for a latent code vector \mathbf{c} that results in the smallest differences between the growth time values (g_k) of the annotated year rings and the network output growth time values $f_\theta(\mathbf{c}, \mathbf{p}_k)$ of the same points. However, the network models the GTF of a larger portion of a tree trunk rather than the block sample (Section 3.2.1). Therefore, it is necessary to also optimize a transformation ($\mathcal{T} = (\mathbf{t}, \mathbf{M}, s)$) that represents the cutout location ($\mathbf{t} \in R^3$), orientation ($\mathbf{M} \in R^{3 \times 3}$), and relative scale ($s \in R$) of the wood block in the tree trunk portion (Figure 6). The latent vector \mathbf{c} and affine transformation ($\hat{\mathcal{T}}$) are estimated by optimizing

$$\arg \min_{\mathbf{c}, \mathcal{T}, a} \sum_k \|f_\theta(\mathbf{c}, \gamma(\mathcal{T}(\mathbf{x}_k))) - (g_k - a)\|_1 + w_{\text{reg}} \|\mathbf{c}\|_2, \quad (5)$$

where $a \in \mathbb{N}$ is an integer value representing an age offset.

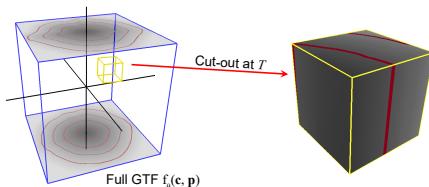


Figure 6: Inferring a GTF. We optimize the latent code \mathbf{c} and cutout transformation \mathcal{T} .

We optimize Equation 5 in two steps. First, we fix $\mathbf{c} = \mathbf{0}$ and perform a coarse grid search to obtain $\mathbf{t}_0, \mathbf{m}_0, s_0$, and a that minimize Equation 5. Next, we initialize the optimization with $\mathbf{c} = \mathbf{0}, \mathbf{t}_0, \mathbf{m}_0$, and s_0 and then iteratively optimize Equation 5 while keeping the age offset a fixed. We show an example of visual outputs during the optimization process in Figure 7.

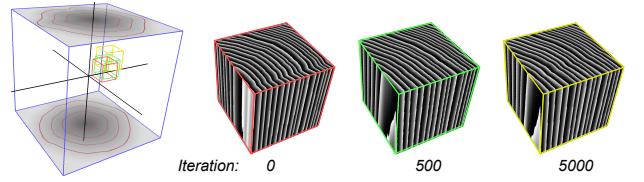


Figure 7: Optimization process. Starting from the coarse grid search result (cube), we iteratively optimize the cutout transformation and latent code. (Note that the output is visualized as an ARF here, see Section 3.2.4).

3.2.4. ARF Conversion

To visualize the periodic pattern of the annual rings, we convert the GTF to an ARF (annual ring field) using the following function:

$$ARF(\mathbf{p}) = (GTF(\mathbf{p}) \bmod 1.0)^2, \quad (6)$$

where mod is the modulo operator. The result is an ARF that has a maximum value at a point where the GTF has a whole value (1.0, 2.0, 3.0, etc., Figure 8). Squaring the output results in elongating the earlywood and shortening the latewood period within an annual layer, which more closely resembles the transition as we observe it in real wood samples, compared to not squaring the output.

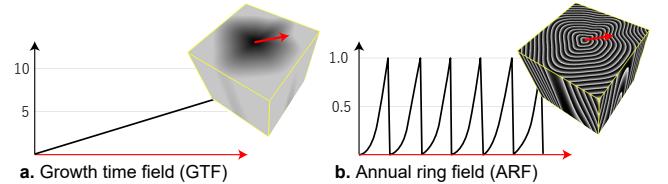


Figure 8: GTF and ARF. The charts plot the values of GTF (a) and ARF (b) along the red arrow.

3.2.5. Training Data for GTF Inference

We procedurally generate various GTFs to train our generic model described in [Section 3.2.1](#). We define a GTF in 2D as

$$GTF(x, y) = \frac{1}{r(\phi)} \sqrt{(x^2 + y^2)}, \quad (7)$$

where ϕ is the angle between vector $(0,1)$ and (x,y) , while $r(\phi)$ is the radial scaling function at the angle ϕ . This formula is based on the parametrization of structural timber [[GBGB95](#)]. The variation in GTF can be obtained by varying $r(\phi)$. We extend the 2D GTF to 3D by defining two radius functions on the planes $z = -1$ and $z = 1$ and linearly interpolating them:

$$GTF(x, y, z) = \frac{1}{r'(\phi, z)} \sqrt{(x^2 + y^2)} \quad (8)$$

$$r'(\phi, z) = \left(1 - \frac{z+1}{2}\right) r_{-1}(\phi) + \frac{z+1}{2} r_1(\phi), \quad (9)$$

where $r_{-1}(\phi)$ and $r_1(\phi)$ are radial scaling functions at $z = -1$ and $z = 1$, respectively. This model assumes a variation of growth speed with the outward direction. For a single direction, we assume constant growing speed across years. Moreover, we assume a straight (non-curved) pith. These simplifications are reasonable when it comes to the study of a smaller block cut out from a portion of a tree stem.

To obtain diverse GTFs, various $r(\phi)$ are required ([Figure 9](#)). We equally sample N_c points from the range $[0, 2\pi]$ and constrain the radius values as $r(\frac{2\pi}{N_c} i) = r_i^c$, where $i = 0, 1, \dots, N_c - 1$. We interpolate between the constraints using Laplacian smoothing to obtain a smooth radius function. We set $N_c = 15$ and determine constraint values r_i^c by randomly sampling from $\mathcal{N}(1.0, 0.1)$. For training our GTF model, we procedurally generated $N = 10,000$ GTFs and sampled $M = 1,600$ points for each of them. Among these sampled points, we sampled 20% of them from the areas close to pith $[-0.1, 0.1] \times [-0.1, 0.1] \times [-1.0, 1.0]$ and the remainder from the entire domain $[-1, 1] \times [-1, 1] \times [-1, 1]$. We use this adaptive sampling because the GTF varies more sharply around the pith.

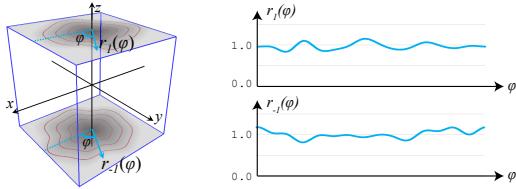


Figure 9: Procedural generation of GTF. We generate a 3D GTF (left) from two radial scaling functions $r_{-1}(\phi)$ and $r_1(\phi)$.

3.3. Local Texture Synthesis

To synthesize the anisotropic local appearance of wood, we propose an orientation-dependent wood texture rendering method using a style transfer technique. To reproduce the anisotropic appearance of wood, we build three color volumes for the XY-, YZ-, and ZX-cross-sections independently using corresponding style transfer models, training a model for each axis orientation ([Section 3.3.1](#)). During rendering, we combine the three volumes depending on the normal orientation of a point on the cut surface ([Section 3.3.2](#)).

3.3.1. Style Transfer for Axis-aligned Cross-sections

The output of the previous stage—global structure inference—is the volumetric ARF data. Based on this data and photographs of the exterior surfaces, we aim to output a the whole volume with a photorealistic appearance. We achieve this by a patch-based style transfer technique, which is capable learning from just one image-pair [[TFK*20](#)]. The style transfer model can thus be retrained from scratch for the unique wood appearance in each color photograph and its corresponding ARF image. The training data consists of many image patches cropped out from the same location of the two images in the pair.

For the purpose of reproducing the anisotropic appearance of wood, we build a colored volume for each of the three orthogonal cross-sections. First, for each side of the cube, a 2D GTF is generated from the annual ring annotations by 2D RBF interpolation. Second, the 2D GTF is converted to 2D ARF, and a style transfer network is trained using the 2D ARF and the corresponding photograph of the side ([Figure 10a](#)). Finally, each of the three style transfer networks is applied to corresponding cross-sections of the inferred 3D ARF in a given resolution, resulting in three volumetric textures (RGB_{XY}, RGB_{YZ}, and RGB_{ZX}; [Figure 10b](#)) to be used in [Equation 10](#) below. Specifically, we use a patch-based style transfer network presented by Texler et al. [[TFK*20](#)], which is capable of learning a translation mapping between two images. It is based on a network architecture that integrates U-net and ResNet (for details, refer to Texler et al. [[TFK*20](#)]).

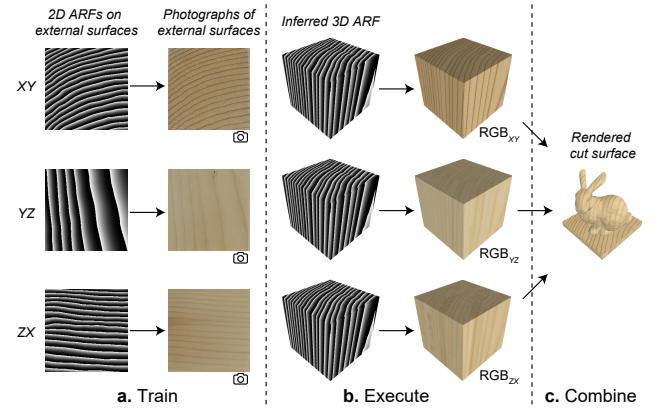


Figure 10: Orientation-dependent style transfer. a) For each of three cross-section orientations (XY, YZ, and ZX), we train a network to translate from a 2D ARF on the external surface to a photograph of the same surface. b) Then, we execute the trained models to translate from a predicted ARF volume to RGB volumes. c) Finally, we render a cut surface inside the block by combining the three RGB volumes.

3.3.2. Orientation-dependent Rendering

We combine the three volumetric textures to render a given cut surface ([Figure 10c](#)) in an orientation-dependent manner. Specifically, We compute the color of each point p on the cut surface as follows:

$$\text{RGB} = \frac{|n_z| \text{RGB}_{XY} + |n_x| \text{RGB}_{YZ} + |n_y| \text{RGB}_{ZX}}{|n_x| + |n_y| + |n_z|}, \quad (10)$$

where RGB is the output color, and $\mathbf{n} = (n_x, n_y, n_z)$ is a normal vector at a surface point \mathbf{p} . [Equation 10](#) is designed based on the assumption that interpolation different directions reproduces the appearance of in-between directions, which is a rough simplification of reality. The color volumes corresponding to the XY-, YZ-, and ZX-planes are denoted by RGB_{XY} , RGB_{YZ} , and RGB_{ZX} , respectively.

4. Results and Evaluation

4.1. Implementation Details

The generic GTF model and style transfer models for orientation-dependent rendering were implemented using PyTorch [[PGM*19](#)] and a computer with an Intel Core i7 8700K CPU and GeForce GTX1080ti SLI. For simplicity, we limited the shape of the input wood block to a cube shape. In total, the time for producing the volumetric textures of a wood block, including the GTF inference and style transfer procedures, was approximately 60 minutes (for details, refer to [Section 4.1.1](#) and [Section 4.1.2](#) below). However, after texture generation, it is possible to render a surface model at a real-time rate using [Equation 10](#).

4.1.1. Global Structure Inference Implementation

It took approximately 8.5 hours to train the generic GTF model for 2000 epochs. This model only needs to be trained once. Regarding GTF inference for a wood exemplar, it took approximately 5 minutes to manually trace the annual rings and approximately one minute to optimize the latent vector \mathbf{z} and cutout transformation.

4.1.2. Local Texture Synthesis Implementation

The patch-based image style transfer model is trained from scratch each time the method is applied to a new image pair (an ARF image and a color photograph). Thus, for each input wood cube sample, we train three networks (one for each of the three cross-section-orientations). We set the resolution of the output wood cube to $260 \times 260 \times 260$. The training data for each network consisted of $N_p = 2,000$ patches, each of size 36×36 . With these settings, it took approximately 15 min to train a network for 300 epoches and three minutes to execute the style transfer on the volume, i.e., to apply the style transfer to each layer of the cube in the given resolution (256).

4.2. Qualitative Evaluation based on Physical Wood Samples

For visual evaluation of our outputs, we generated volumetric images of several physical wood samples and visually compared rendered images and photographs ([Figure 1](#), [Figure 11](#)). We applied our method to a multitude of samples including various common hard- and softwood species, such as oak and pine ([Figure 11](#)). We prepared samples by photographing their external surfaces and manually tracing the annual rings. We then applied our method to generate volumetric color textures and rendered Stanford bunny models. In addition, we physically made various cuts of the wood samples to visually compare the revealed internal textures to those inferred by our method. Planar cuts were made with a saw ([Figure 11](#), wood samples 1-5), while 3D geometries were fabricated using a 3-axis CNC milling machine ([Figure 11](#), wood samples 6-7).

We also fabricated a relatively more complex geometry using a

CNC Lathe machine ([Figure 1](#)). Comparing the rendered prediction ([Figure 1e](#)) and the fabricated artifact ([Figure 1f](#)), we observe a high resemblance in the location of the the annual ring lines, while there are discrepancies in local appearance (color, contrast, etc.). This can be partially explained by the fact that the fabricated artifact is polished and therefore has a smoother surface finish compared to the original wood block from which the prediction is made. In hindsight, it would have been better to not polish the artifact, or alternatively, to treat the surface of the original wood block in the same way as we intended to treat the cut surface. Other explanations of the discrepancies, apart from inaccuracies of our inference method, are differences between the physical and rendered lighting conditions, which we did not meticulously control in our experimental setup.

4.3. Quantitative Evaluation based on Physical Wood Samples

Based on the planar cut wood samples from the qualitative evaluation ([Figure 11](#), wood samples 1-5), we conducted a quantitative ablation and baseline comparison study to evaluate the accuracy and robustness of our global structure inference method. We photographed the cut surface and manually traced their annual rings while labeling them by year to obtain a ground truth ([Figure 12-left](#)). Then we compared inferred annual ring patterns to the ground truth by measuring what percentage of the points on the inferred annual rings is within two different thresholds (1.0 and 2.0 mm) from the ground truth. We performed this evaluation on outputs created from full annotations, i.e., complete traces of annual rings of all exterior sides, ([Figure 12-middle](#)) and partial annotations, i.e., traces of annual rings on half of the block only ([Figure 12-right](#)). We also compare the outputs of our method to a baseline method of naïve radial basis function (RBF) interpolation. We calculate the averages for each experiment, and compute probability values (p-values) using the paired t-test [[Stu08](#)] with two-tailed distribution.

The results show that our method always has a higher average accuracy ([Table 1](#)). However, the results are not statistically significant (p-values > 0.05), indicating that a higher number of samples are needed before drawing definite conclusions. That being said, our method performed best in comparison to the baseline for the partial annotations, which suggests that our method might be comparatively robust against partial constraints, which is important because manually drawing fewer strokes is easier compared to tracing all annual rings on the exterior surface photographs.

Furthermore, as for computations times, inferring the GTF with our method took one minute on average, while RBF interpolation took three minutes. Note that these are the computation times of the first step of global inference only; they do not include the second step (style transfer), which would add to the computation times. Finally, we assume that further acceleration is possible because the current implementation was done naively using PyTorch [[PGM*19](#)].

4.4. Further Experiments

A side-by-side comparison between images produced based on our orientation-dependent and orientation-agnostic renderings shows that our method better captures the differences between surface orientations while also producing more natural-looking color variations ([Figure 13](#)). We also show the effect of varying the orientation

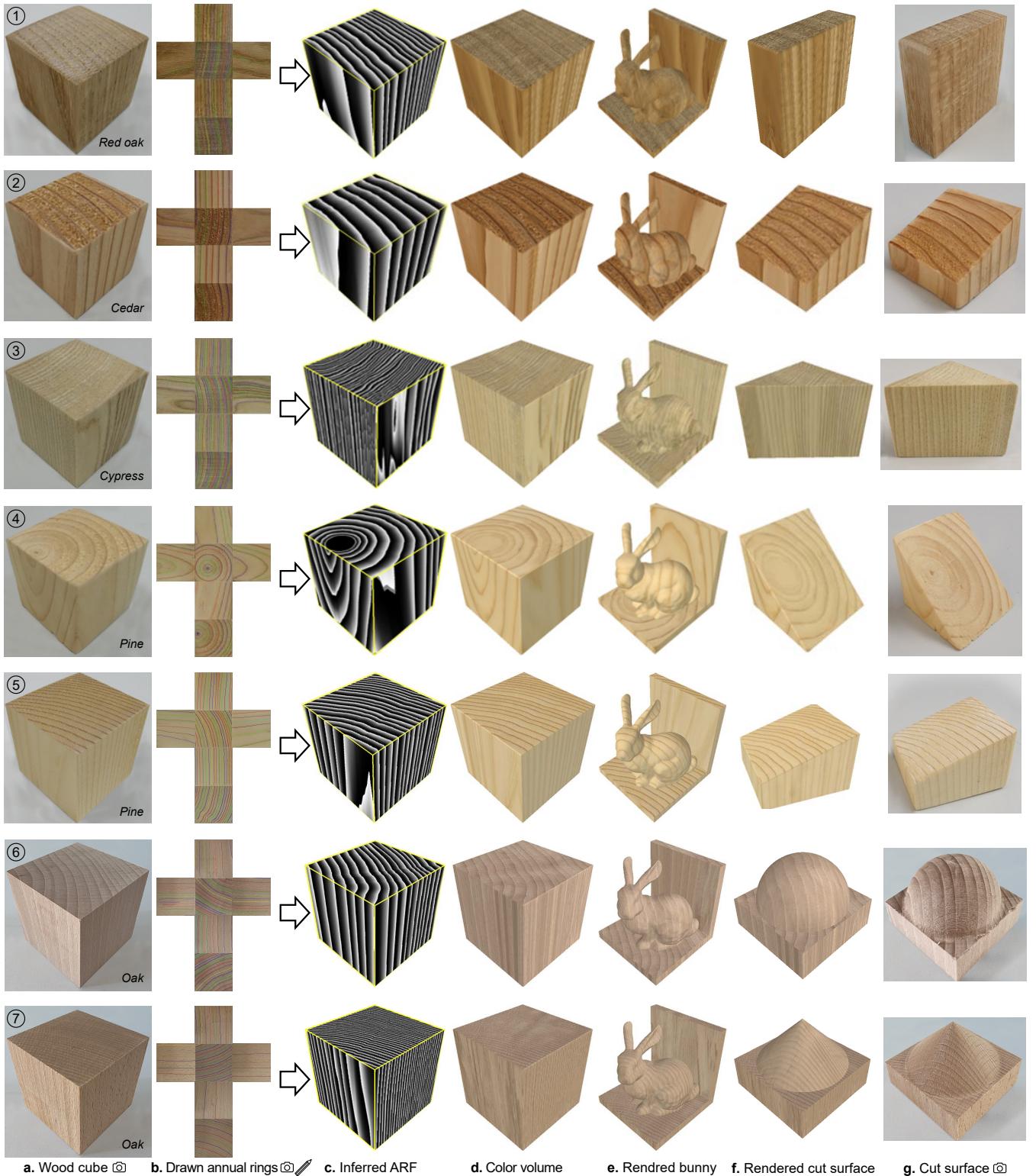


Figure 11: Photographs of five sample wood cubes and volumetric textures generated using our method. Each row shows a photograph of a) a wood cube exemplar, b) its six external surfaces with annotations, c) the inferred ARF, d) the color volume after applying style transfer, e) a rendered Stanford bunny inside the block, f) a rendered cut surface and g) a photograph of the corresponding physically cut surface.

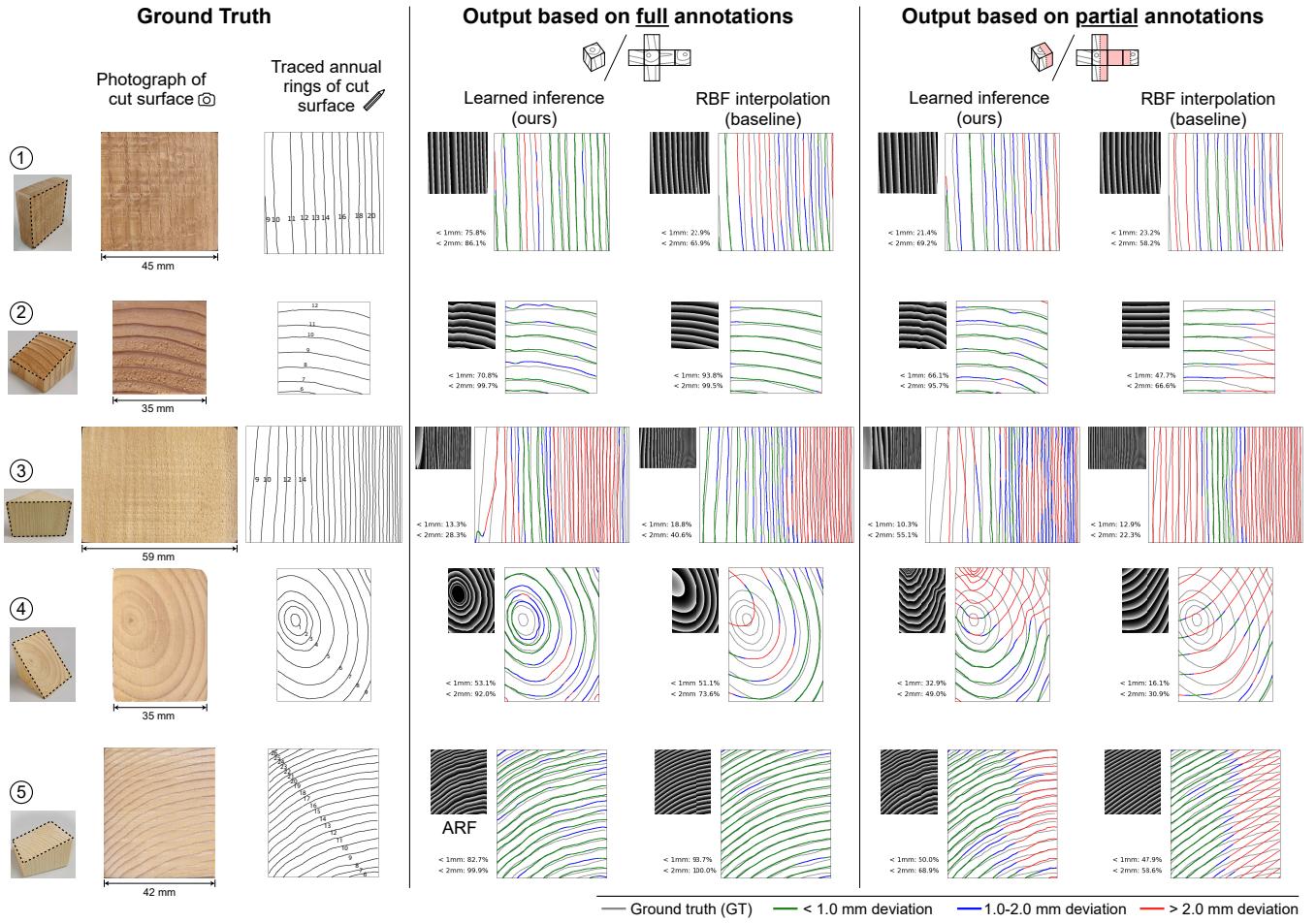


Figure 12: Quantitative ablation and baseline comparison study. Left: the ground truth annual ring pattern of cut surfaces of physical wood samples. Middle: inferred results based on full annotations. Right: inferred results based on partial annotations. For a quantitative compilation, refer to Table 1.

Table 1: Quantitative results measuring the percentage of predicted annual ring lines on a cut surface within different thresholds (1.0 mm and 2.0 mm) of the ground truth (refer to Figure 12).

Wood Sample #	Full annotation				Partial annotation			
	1 mm threshold		2 mm threshold		1 mm threshold		2 mm threshold	
	Ours (%)	Baseline (%)	Ours (%)	Baseline (%)	Ours (%)	Baseline (%)	Ours (%)	Baseline (%)
1	75.8	22.9	86.1	65.9	21.4	23.2	69.2	58.2
2	70.8	93.8	99.7	99.5	66.1	47.7	95.7	66.6
3	13.3	18.8	28.3	40.6	10.3	12.9	55.1	22.3
4	53.1	51.1	92.0	73.6	32.9	16.1	49.0	30.9
5	82.7	93.7	99.9	100.0	50.0	47.9	68.9	58.6
Averages	59.1	56.1	81.2	75.9	36.1	29.6	67.6	47.3
p-values		0.88		0.77		0.61		0.13

of a cut surface relative to the material (Figure 14) and what happens when we apply different local appearances to the same global structure (Figure 15).

5. Limitations and Future Work

5.1. Input constraints and requirements

Our current method requires an orthogonal cuboid-shaped wood exemplar. However, as for method stage one (global structure in-

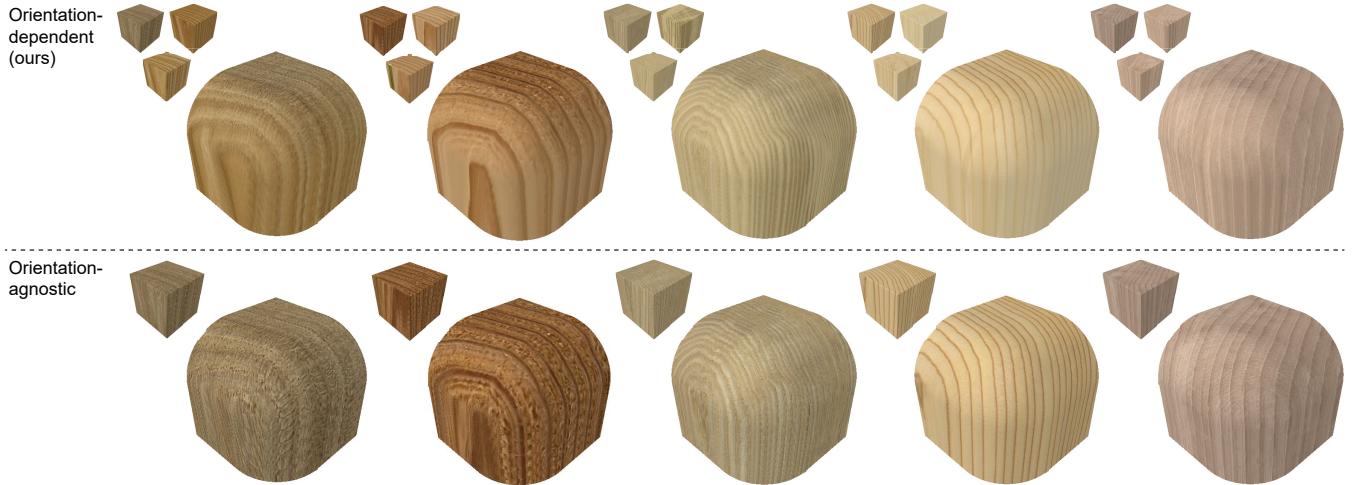


Figure 13: Top: Orientation-dependent rendering based-on three color volumes constructed from the XY, YZ, and ZX cross sections, respectively, and mixed based on the normal direction of a point on the cut surface. Bottom: Orientation-agnostic rendering based on one color volume constructed from the XY cross-section.

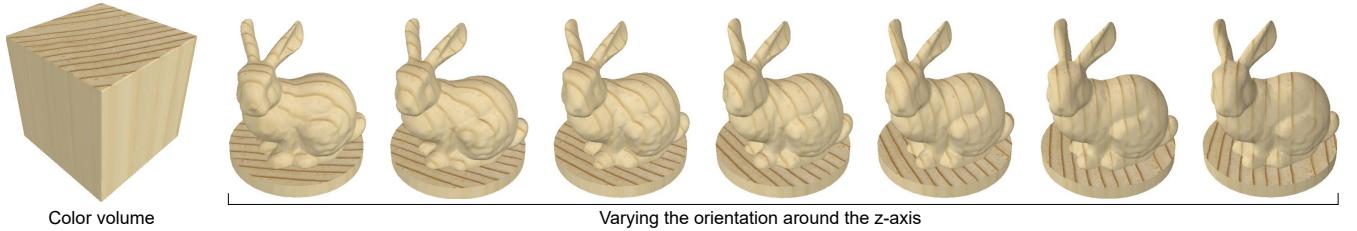


Figure 14: The cut surface is rotated incrementally around the z-axis relative to the material by 15 degrees. This changes the annual ring pattern on the surface.

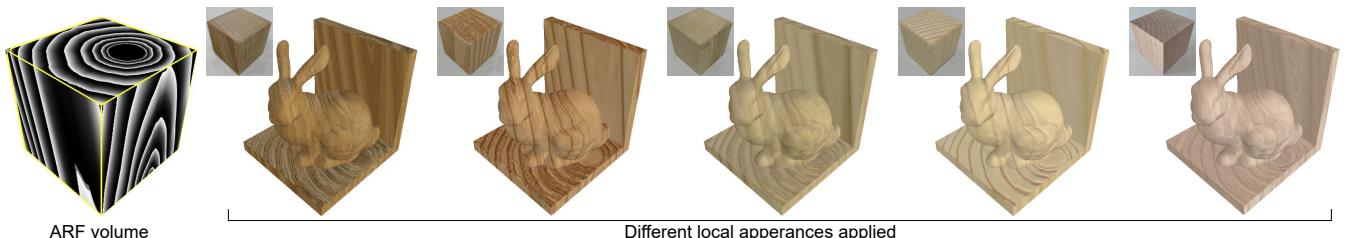


Figure 15: Cut surfaces with the same ARF (global structure) but different local appearances. These cross-combinations are possible because of the separation of the two steps of the method—global structure inference and local texture synthesis.

ference), the inference is not theoretically restricted to points on planar surfaces (the exterior surfaces of the cuboid) because the GTF model takes any point in 3D as input and produces a growth time value. That being said, annotating annual rings on arbitrary shapes would be more challenging as it would require mapping the texture of the 3D shape to a 2D surface or, alternatively, a 3D annotation interface. Extending stage two of our method (local texture synthesis) to arbitrary shapes can be challenging because our current orientation-dependent rendering method assumes that we can obtain the appearance of three orthogonal planes of a block shape. In the future, we plan to extend our method to support arbitrary shapes.

Another limitation is that our method requires manually annotated annual rings for reconstructing the GTF. It is robust against incomplete annotations (refer to Section 4.3), but nevertheless, we plan to develop an automatic annual ring extraction method based on style transfer and edge extraction techniques.

5.2. Quality of output: Global structure inference

Our method cannot generate large-scale distortions caused by the grafting of the tree, such as knots (Figure 16a). It also does not reconstruct small-scale distortions of the GTF appearing on an ex-

ternal surface (Figure 16b). Owing to this limitation, the small-scale distortions of the ARFs of all our outputs look similar in terms of level of noise and smoothness. This is partially due to our training data. Therefore, we plan to improve the accuracy and diversity of the ARF reconstruction by increasing the training dataset diversity. Moreover, as for evaluation of the inferred pattern, our quantitative study (Section 4.3) is based on five samples from four different wood species. The results are not statistically significant, indicating that it would be desirable to create a dataset of a larger quantity of samples to enable more robust evaluation. It would also be preferable to test our system on a larger number of different wood species.

5.3. Quality of output: Local texture synthesis

Our rendered prediction is made with a simple reflection model (ambient and diffuse light), resulting in a more unobstructed and clearly visible texture pattern, while the photographs of the artifacts sometimes have additional real-world effects such as shadows, focus blur, and highlights. A more advanced rendering model could solve some of these issues in the future. Moreover, although rare, the style transfer produces unsatisfactory results when the annual rings on an external surface are too sparse (Figure 17). This is because the style transfer is trained using small patches of the input images, and some of the patches do not contain any annual rings if they are too sparse. Further, it is computationally costly to train multiple style transfer models for each sample. Therefore, we plan to investigate how a single conditional style transfer model for each species of wood can be learned and used for the reconstruction of the local appearance. On the one hand, this could accelerate the style transfer process and mitigate failure cases caused by too few rings (Figure 17). On the other hand, it might yield poorer results when applied to wood samples that significantly deviate from the appearance of those in the training dataset.

5.4. Future applications

It is common to decompose complex models into multiple parts due to constraints in CNC fabrication of wood artifacts [MLS*18,

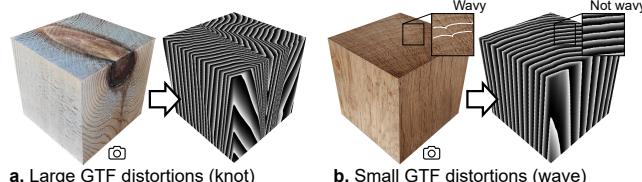


Figure 16: Failure cases. Our method does not succeed in reproducing a) a very large or b) very small distortions of the GTF.



Figure 17: Failure case. The style transfer gives unsatisfactory outputs when the annual rings are too sparse on an external surface. The white box indicates a patch without structural information.

HMA15]. Inferring the internal pattern using our method can help systems in matching the texture of the parts for improved appearance of the assembled artifact. This is similar to previous systems matching patterns over garment seams [WSH19]. Another possible future application is to complement shape completion with material texture completion for cases when there is an incomplete 3D scan of a solid wood object. Moreover, previous work proposed a method for strength-optimization of the orientation of artifacts fabricated using Fused Deposition Modeling (FDM) 3D printing, leveraging the anisotropic property of the filament bounds [US13]. A similar technique could be applied to increase the strength of wood artifacts fabricated by CNC milling or similar methods, after obtaining a model of its volumetric structure.

6. Conclusion

In this paper, we introduced a new research problem of inferring the volumetric texture of solid wood based on photographs of the visible exterior surfaces. We addressed this problem by a two-stage method. First, we employed a learned model for inferring the volumetric global structure. We demonstrated that this model is marginally more accurate than a baseline of basic interpolation under full annotations and substantially better under partial annotations, while being three times faster. Second, we proposed a novel orientation-dependent rendering method that replicates the anisotropic appearance of the local features and colors. The proposed system is useful for physical applications—such as subtractive manufacturing—and also for virtual ones—such as object breaking.

7. Acknowledgements

This work was supported by a collaborative research fund between Mercari Inc. R4D and RIISE, University of Tokyo, JST ACT-X Grant Number JPMJAX210P, Japan, JSPS KAKENHI Grant Number JP23K19994, JSPS Grant-in-Aid JP23K16921, and JST Ad-CORP, Grant Number JPMJKB2302, Japan. We would further like to thank the anonymous reviewers for their constructive feedback.

8. Conflict of Interest Statement

We have no conflicts of interest to disclose.

9. Data Availability Statement

The data in Table 1 are available in the supplementary material of this article. Further data are available from the corresponding author upon reasonable request.

References

- [CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 4
- [DLTD08] DONG Y., LEFEBVRE S., TONG X., DRETTAKIS G.: Lazy solid texture synthesis. In *Proceedings of the Nineteenth Eurographics conference on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2008), EGSR’08, Eurographics Association, pp. 1165–1174. doi:10.1111/j.1467-8659.2008.01254.x. 3

- [GBGB95] GRÖNLUND A., BJÖRKLUND L., GRUNDBERG S., BERGGREN G.: *Manual för furustambank*. Tech. rep., Luleå Technical University., 1995. 6
- [GHS*22] GUERRERO P., HAŠAN M., SUNKAVALLI K., MĚCH R., BOUBEKEUR T., MITRA N. J.: Matformer: A generative model for procedural materials. *ACM Trans. Graph.* 41, 4 (jul 2022). URL: <https://doi.org/10.1145/3528223.3530173>, doi: 10.1145/3528223.3530173. 3
- [GJB*20] GUO J., JIANG H., BENES B., DEUSSEN O., ZHANG X., LISCHINSKI D., HUANG H.: Inverse procedural modeling of branching structures by inferring l-systems. *ACM Trans. Graph.* 39, 5 (jun 2020). URL: <https://doi.org/10.1145/3394105>, doi: 10.1145/3394105. 3
- [GRGH18] GUTIERREZ J., RABIN J., GALERNE B., HURTUT T.: On Demand Solid Texture Synthesis Using Deep 3D Networks. working paper or preprint, Dec. 2018. URL: <https://hal.archives-ouvertes.fr/hal-01678122>. 3
- [HDR19] HU Y., DORSEY J., RUSHMEIER H.: A novel framework for inverse procedural texture modeling. *ACM Trans. Graph.* 38, 6 (nov 2019). URL: <https://doi.org/10.1145/3355089.3356516>, doi: 10.1145/3355089.3356516. 3
- [HGH*22] HU Y., GUERRERO P., HASAN M., RUSHMEIER H., DESCHAINTRE V.: Node graph optimization using differentiable proxies. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. URL: <https://doi.org/10.1145/3528233.3530733>, doi: 10.1145/3528233.3530733. 3
- [HHD*22] HU Y., HE C., DESCHAINTRE V., DORSEY J., RUSHMEIER H.: An inverse procedural modeling pipeline for svbrdf maps. *ACM Trans. Graph.* 41, 2 (jan 2022). URL: <https://doi.org/10.1145/3502431>, doi: 10.1145/3502431. 3
- [HMA15] HERHOLZ P., MATUSIK W., ALEXA M.: Approximating free-form geometry with height fields for manufacturing. *Comput. Graph. Forum* 34, 2 (may 2015), 239–251. URL: <https://doi.org/10.1111/cgf.12556>, doi: 10.1111/cgf.12556. 11
- [HMR19] HENZLER P., MITRA N. J., RITSCHEL T.: Learning a neural 3d texture space from 2d exemplars. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 3
- [IOI06] IJIRI T., OWADA S., IGARASHI T.: The sketch L-system: Global control of tree modeling using free-form strokes. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2006), vol. 4073 LNCS. doi: 10.1007/11795018__13. 3
- [JSM*20] JIANG C., SUD A., MAKADIA A., HUANG J., NIESSNER M., FUNKHOUSER T.: Local implicit grid representations for 3d scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). 4
- [KBI14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 4
- [KEB05] KWATRA V., ESSA I., BOBICK A., KWATRA N.: Texture optimization for example-based synthesis. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, Association for Computing Machinery, p. 795–802. URL: <https://doi.org/10.1145/1186822.1073263>, doi: 10.1145/1186822.1073263. 3
- [KFCO*07] KOPF J., FU C.-W., COHEN-OR D., DEUSSEN O., LISCHINSKI D., WONG T.-T.: Solid texture synthesis from 2d exemplars. *ACM Trans. Graph.* 26, 3 (jul 2007), 2–es. URL: <https://doi.org/10.1145/1276377.1276380>, doi: 10.1145/1276377.1276380. 3
- [KSG*15] KRATT J., SPICKER M., GUAYAQUIL A., FISER M., PIRK S., DEUSSEN O., HART J. C., BENES B.: Woodification: User-Controlled Cambial Growth Modeling. *Computer Graphics Forum* (2015). doi: 10.1111/cgf.12566. 2, 3
- [LDHM16] LIU A. J., DONG Z., HAŠAN M., MARSCHNER S.: Simulating the structure and texture of solid wood. *ACM Trans. Graph.* 35, 6 (nov 2016). URL: <https://doi.org/10.1145/2980179.2980255>, doi: 10.1145/2980179.2980255. 3
- [Lin68] LINDENMAYER A.: Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology* 18, 3 (1968). doi: 10.1016/0022-5193(68)90079-9. 3
- [LIY*22] LARSSON M., IJIRI T., YOSHIDA H., HUBER J. A. J., FREDRIKSSON M., BROMAN O., IGARASHI T.: Procedural texturing of solid wood with knots. *ACM Trans. Graph.* 41, 4 (jul 2022). URL: <https://doi.org/10.1145/3528223.3530081>, doi: 10.1145/3528223.3530081. 2, 3
- [LP00] LEFEBVRE L., POULIN P.: Analysis and synthesis of structural textures. In *Graphics Interface 2000* (May 2000), pp. 77–86. 3
- [MLS*18] MUNTONI A., LIVESU M., SCATENI R., SHEFFER A., PANIZZO D.: Axis-aligned height-field block decomposition of 3d shapes. *ACM Trans. Graph.* 37, 5 (oct 2018). URL: <https://doi.org/10.1145/3204458>, doi: 10.1145/3204458. 11
- [MON*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 4
- [MPW06] MANN J., PLANK M., WILKINS A.: Tree growth and wood formation — applications of anisotropic surface growth. In *Proceedings of the Mathematics in Industry Study Group* (2006), pp. 153–192. 2, 3
- [MRS*20] MARTIN-BRUALLA R., RADWAN N., SAJJADI M. S. M., BARRON J. T., DOSOVITSKIY A., DUCKWORTH D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. *CoRR abs/2008.02268* (2020). URL: <https://arxiv.org/abs/2008.02268>, arXiv: 2008.02268. 4
- [MST*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I* (Berlin, Heidelberg, 2020), Springer-Verlag, p. 405–421. URL: https://doi.org/10.1007/978-3-030-58452-8_24, doi: 10.1007/978-3-030-58452-8_24. 4
- [MWAM05] MARSCHNER S. R., WESTIN S. H., ARBREE A., MOON J. T.: Measuring and modeling the appearance of finished wood. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, Association for Computing Machinery, p. 727–734. URL: <https://doi.org/10.1145/1186822.1073254>, doi: 10.1145/1186822.1073254. 3
- [OMN*19] OECHSLE M., MESCHEDER L. M., NIEMEYER M., STRAUSS T., GEIGER A.: Texture fields: Learning texture representations in function space. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 4530–4539. 3
- [PABG20] PORTEMIER T., ARJOMAND BIGDELI S., GOKSEL O.: Gramgan: Deep 3d texture synthesis from 2d exemplars. In *Advances in Neural Information Processing Systems* (2020), Larochelle H., Ranzato M., Hadsell R., Balcan M., Lin H., (Eds.), vol. 33, Curran Associates, Inc., pp. 6994–7004. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4df5bde009073d3ef60da64d736724d6-Paper.pdf. 3
- [PCOS10] PIETRONI N., CIGNONI P., OTADUY M., SCOPIGNO R.: Solid-texture synthesis: A survey. *IEEE Computer Graphics and Applications* 30, 4 (2010), 74–89. doi: 10.1109/MCG.2009.153. 3
- [PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVE-GROVE S.: Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2019), vol. 2019-June. doi: 10.1109/CVPR.2019.00025. 2, 4

- [PGM*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S.: Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. 7
- [POB*07] PIETRONI N., OTADUY M. A., BICKEL B., GANOVELLI F., GROSS M.: Texturing Internal Surfaces from a Few Cross Sections. *Computer Graphics Forum* (2007). doi:[10.1111/j.1467-8659.2007.01087.x](https://doi.org/10.1111/j.1467-8659.2007.01087.x). 3
- [SPH11] SELLIER D., PLANK M. J., HARRINGTON J. J.: A mathematical framework for modelling cambial surface evolution using a level set method. *Annals of Botany* 108, 6 (2011). doi:[10.1093/aob/mcr067](https://doi.org/10.1093/aob/mcr067). 2
- [Stu08] STUDENT: The probable error of a mean. *Biometrika* (1908), 1–25. 7
- [TFK*20] TEXLER O., FUTSCHIK D., KUČERA M., JAMRIŠKA O., SOCHOROVÁ V., CHAI M., TULYAKOV S., SYÝKORA D.: Interactive style transfer to live video streams. SIGGRAPH '20, Association for Computing Machinery. URL: <https://doi.org/10.1145/3407662.3407752>, doi:[10.1145/3407662.3407752](https://doi.org/10.1145/3407662.3407752). 2, 6
- [US13] UMETANI N., SCHMIDT R.: Cross-sectional structural analysis for 3d printing optimization. In *SIGGRAPH Asia 2013 Technical Briefs* (New York, NY, USA, 2013), SA '13, Association for Computing Machinery. URL: <https://doi.org/10.1145/2542355.2542361>, doi:[10.1145/2542355.2542361](https://doi.org/10.1145/2542355.2542361). 11
- [WSH19] WOLFF K., SORKINE-HORNUNG O.: Wallpaper pattern alignment along garment seams. *ACM Trans. Graph.* 38, 4 (jul 2019). URL: <https://doi.org/10.1145/3306346.3322991>, doi:[10.1145/3306346.3322991](https://doi.org/10.1145/3306346.3322991). 11
- [XWC*19] XU Q., WANG W., CEYLAN D., MECH R., NEUMANN U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems* (2019), Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., (Eds.), vol. 32, Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2019/file/39059724f73a9969845dfe4146c5660e-Paper.pdf>. 4
- [ZWG*21] ZHAO X., WANG L., GUO J., YANG B., ZHENG J., LI F.: Solid texture synthesis using generative adversarial networks. *CoRR abs/2102.03973* (2021). URL: <https://arxiv.org/abs/2102.03973>, arXiv:2102.03973. 3